# 講義ノート6：
# 動的計画法とモデルの再帰的表現

蓮見 亮

2013年2月21日

## 目次

ここまで最適成長モデル、RBC モデル、ニューケインジアン・モデルなどの動学的一般均衡モデルを紹介してきたが、その解き方としてラグランジュの未定乗数法により連立方程式を導出し、遠い将来に定常状態に至るという仮定の下で解くという方法を採用してきた。主としてこの方法が最も数学的に簡単であるという理由からだが（方法の比較は 5 節参照）、ここで問題となるのはなぜ「遠い将来に定常状態に至る」と仮定してよいかである。実際、これまで紹介してきたの動学的一般均衡モデルではその仮定は妥当であるのだが、それを議論するにはラグランジュの未定乗数法に代えてベルマン方程式という式を用いた最適化手法（動的計画法）を用いる必要がある。

# 1　ベルマン方程式を用いた最適化手法

## 1.1　ベルマン方程式

ベルマン方程式とは、以下のような離散時間の動的計画問題

$$\begin{aligned} &\max_{\boldsymbol{x}} \quad \sum_{i=0}^{\infty} \beta^i U(x_{t+i}) \\ &\text{s.t.} \quad s_{i+1} = g(s_i, x_i),\ i \geq t \end{aligned} \tag{1}$$

を解く際に用いる方程式をいう。ここでは $x_t, s_t$ はベクトルでなくそれぞれ 1 変数とする。$x_t$ がジャンプ変数、$s_t$ が状態変数にあたる[*1]。

ノート 2 以降これらの用語は既に用いているが、改めて整理しておこう。一般に、最適化モデルに含まれる変数は内生変数と外生変数に区別でき、内生変数はさらにジャンプ変数と状態変数に区別できる。

$$\begin{array}{ll} \text{内生変数} & \begin{cases} \text{ジャンプ変数 } x_t \\ \text{状態変数 } s_t \end{cases} \\ \text{外生変数} & \end{array}$$

ジャンプ変数とは、操作変数（control variable）とも呼ばれ、最適化行動しようとする主体が当期の状態に基づく最適化問題の解として毎期決定する変数をいう。状態変数とは、最適化行動しようとする主体が間接的にしかコントロールできないか、あるいは全くコントロールできない変数をいう。状態変数は先決変数とも呼ばれ、前期の状態と行動に基づき先決的に決定される。上記の $s_{t+1} = g(s_t, x_t)$ のように、遷移式の形式で予め与えられるという特徴がある。これに対して、外生変数は内生変数の値を求めるために予め何らかの値を与えておく必要のある変数のことであるが、本章で考える動的計画問題には外生変

[*1] ジャンプ変数、状態変数の一方もしくは両方がベクトルの場合のベルマン方程式は、4.2 節で議論する。

数は含まれないものとする。この場合、シミュレーションは状態変数の初期値を変えることにより行う。

ここで、価値関数 $V(s_t)$ を

$$V(s_t) = \max_{\boldsymbol{x}} \sum_{i=0}^{\infty} \beta^i U(x_{t+i}) \tag{2}$$

と定義する。すなわち、価値関数とは動的計画問題の目的関数のとりうる最大値を状態変数 $s_t$ の関数として表したものである。一般に、外生変数が存在しなければ、動的計画問題の目的関数のとりうる最大値は状態変数の初期値のみによって決まる[*2]。(2) 式は、1 期先の価値関数 $V(s_{t+1}) = V(g(s_t, x_t))$ を用いて、

$$V(s_t) = \max_{x_t} \{U(x_t) + \beta V(g(s_t, x_t))\} \tag{3}$$

と再帰的な形式に書き直すことができ、これをベルマン方程式という[*3]。ベルマン方程式は、数ではなく関数を解とする関数方程式であり、両辺に出てくる関数 $V$ がその解となる関数である。

外生変数が存在しなければ、最適なジャンプ変数 $x_t$ の値は当期の状態 $s_t$ のみによって決まり、そのような関数 $h : S \ni s_t \mapsto x_t \in R$ を行動関数（policy function）と呼ぶ[*4]。つまり、ジャンプ変数のとる値を状態変数の関数として表したものが行動関数で、argmax という数学記号を用いて

$$h(s_t) = \operatorname*{argmax}_{x_t} \{U(x_t) + \beta V(g(s_t, x_t))\} \tag{4}$$

と表せる[*5]。行動関数 $h$ は、

$$V(s_t) = U(h(s_t)) + \beta V(g(s_t, h(s_t))) \tag{5}$$

という関係を満たす。(4) 式もしくは (5) 式の関係を用いて関数 $h$ を求めることもできるが、多くの場合、解析的に求めることは不可能であり、それを具体的に求めたい場合には、関数 $h$ を他の関数で近似する必要がある[*6]。

---

[*2] 最適成長モデルを例に取れば、外生ショックが全く存在しなければ、効用の取りうる最大値は初期の資本ストックの大きさのみによって決まる。

[*3] このように再帰表現できるのは、いま考えている問題が無限期間の動的最適化問題だからである。この場合、価値関数 $V$ の形が時間 $t$ に依存しなくなる。有限期間の動的最適化問題の場合には、ベルマン方程式に現れる価値関数は時間にあらわに依存する。外生変数が存在する場合も同様である。

[*4] $S$ はジャンプ変数の定義域である。実は、policy function という語に対して適切な訳語が見当たらない。政策関数と呼んでもよいが、経済学の文脈だと誤解を招きやすい。policy function とそのまま呼んでしまってもよいが、ここでは仮に行動関数という訳語を当てておく。

[*5] その意味は、他の条件が同じならば、同じ状態が複数回起こったときには同じ行動をとるはずだということである。

[*6] 線型近似やチェビシェフ近似など。詳しくは後述。

## 1.2 ラグランジュの未定乗数法との関係

これまで、(1) のような最適化問題に対して、ラグランジュの未定乗数法により最適化の条件を導出してきたが、ベルマン方程式 (3) を用いても全く同じ式を導出することができる。まず、ベルマン方程式 (3) の右辺の括弧内のジャンプ変数 $x_t$ による微分がゼロに等しいこと、すなわち

$$U'(x_t) + \beta V'(g(s_t, x_t))g_x(s_t, x_t) = 0 \tag{6}$$

が最適化の一階条件である。次に、包絡線定理（Envelope Theorem）と呼ばれる定理から、価値関数 $V(s_t)$ の微分は、ベルマン方程式 (3) の右辺の括弧内の $s_t$ による偏微分に等しいこと、すなわち

$$V'(s_t) = \beta V'(g(s_t, x_t))g_s(s_t, x_t) \tag{7}$$

が成り立つことことが知られている[*7]。ただし、$g_x = \frac{\partial g}{\partial x}, g_s = \frac{\partial g}{\partial s}$ で、合成関数の微分法を用いている。

【例】最適成長モデル（ノート 2）のベルマン方程式とオイラー方程式

最適成長モデルのベルマン方程式は、$A_t = 1$ とすると

$$V(K_t) = \max_{C_t} \{\ln(C_t) + \beta V\left(g(K_t, C_t)\right)\} \tag{8}$$

$$K_{t+1} = g(K_t, C_t) = K_t{}^{\alpha} + (1-\delta)K_t - C_t \tag{9}$$

と書けるが、これに最適化の一階条件 (6) 式と包絡線定理 (7) 式を当てはめると、

$$\frac{1}{C_t} - \beta V'(K_{t+1}) = 0 \tag{10}$$

$$V'(K_t) = \beta V'(K_{t+1})(\alpha K_t^{\alpha-1} + 1 - \delta) \tag{11}$$

という方程式が得られる。(10) 式から $V'(K_{t+1}) = \frac{1}{\beta C_t}, V'(K_t) = \frac{1}{\beta C_{t-1}}$ であることを利用して、(11) 式から $V'$ を消去して整理すると、

$$\frac{C_t}{C_{t-1}} = \beta(\alpha K_t^{\alpha-1} + 1 - \delta) \tag{12}$$

というオイラー方程式が得られる。これは、ノート 2 の (42) 式に $A_t = 1$ を代入し、時間のインデックス $t$ を $t-1$ に置き換えたものと一致する。

---

[*7] ベルマン方程式 (3) の右辺の括弧内で定義した関数は、$x_t = h(s_t)$ で $V(s_t)$ と接することを意味する。どのような場合に包絡線定理が成り立つかなど厳密な議論に関心のある読者は、“Microeconomic Analysis” (文献 [6]) の 27 章、“Recursive Macroeconomic Theory” (文献 [3]) の 4 章、“Introduction to Modern Economic Growth”（文献 [1]）の 6 章など参照していただきたい。

## 1.3　行動関数（policy function）の求め方

ベルマン方程式 (3) の解となる関数 $h : s_t \mapsto x_t$、すなわち行動関数を求める方法はいくつか知られているが、そのうち

- 線型写像で近似した行動関数を行列計算により求める方法
- 関数近似した行動関数を (5) 式の関係を用いて繰り返し計算によって求める方法

の 2 通りを紹介する。

前者の線型写像で近似した行動関数を行列計算により求める方法を用いると、モデルを行列により再帰的に表現できる。具体的には、以下のような手順で計算する。

モデルの再帰的表現

（1）最適化の一階条件を求め、モデルを定常状態周りで対数線型近似する。
（2）定常状態周りで対数線型近似したモデルのジャンプ変数に対応する行動関数が、$p$ を実数として

$$\hat{x}_t = p\hat{s}_t \tag{13}$$

という関数形をとると仮定して $p$ を求める。ただし、$x^*, s^*$ をそれぞれ定常均衡値として $\hat{x}_t = \ln(x_t) - \ln(x^*),\ \hat{s}_t = \ln(s_t) - \ln(s^*)$ を意味する。

ジャンプ変数、状態変数の一方もしくは両方がベクトルの場合は、実数 $p$ が行列 $P$ となる[*8]。最適成長モデルの場合のようにジャンプ変数と状態変数の両方とも 1 変数である場合でも、実数 $p$ を具体的に求める際に、行列の固有値に関する知識を用いたほうが見通しが良い。したがって、この方法の詳細を 3 節（モデルの再帰的表現）で説明する前に 2 節で行列の定義と使い方を紹介する。

後者の関数近似した行動関数を繰り返し計算によって求める方法について、詳しくは 4 節（行動関数の関数近似）で説明するが、先に次の 1.4 節で具体例を紹介する。

## 1.4　なぜ「遠い将来に定常状態に至る」と仮定してよいかについての考察

ノート 2〜5 のモデルを解く際、「遠い将来に定常状態に至る」と仮定してラグランジュの未定乗数法により導出した連立方程式を解いた。この仮定は、最適解の存在条件とも関連しており、厳密な議論は本稿の想定するレベルを大きく超える。したがって、ここでは、1 つの例として、上記の最適成長モデルのベルマン方程式を満たす行動関数を関数近

[*8] 一般に、線型写像は行列で表現できる。

似を用いて求め、この仮定の妥当性を議論しよう。

4 節で説明する方法を用いて近似的に求めた行動関数 $C_t = \tilde{h}(K_t)$ を図示すると、図 1 のようになる[*9]。このとき、価値関数の近似 $\tilde{V}(K_t)$ は、図 2 のように求まる。パラメータは $\alpha = 0.3,\ \beta = 0.99,\ \delta = 0.25$ とおいた。

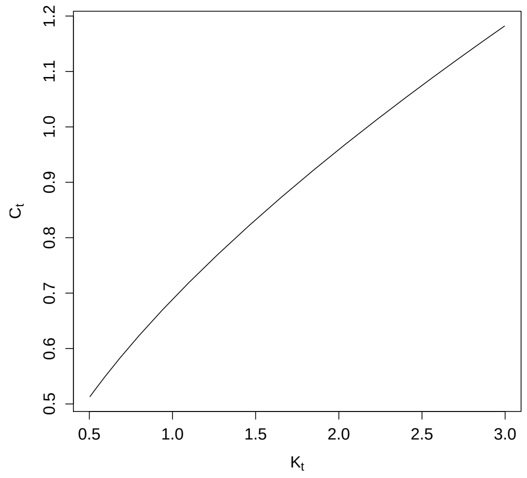


図 1　行動関数 $C_t = \tilde{h}(K_t)$

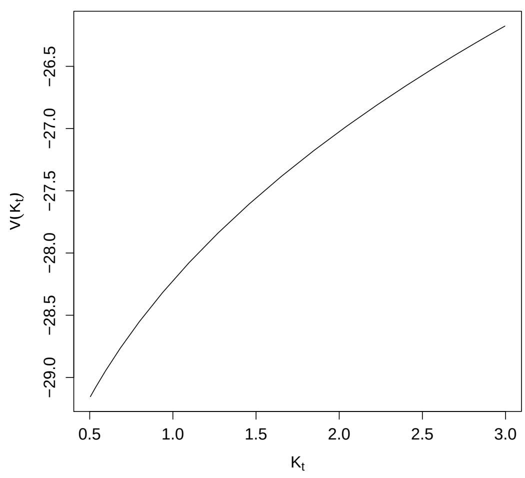


図 2　価値関数 $V(K_t)$

この $\tilde{h}$ を用いて、例えば $K_1 = 0.5K^*$ という初期値を与えてシミュレーションを行う（$K^*$ は定常状態における $K_t$ の値）、すなわち

$$C_t = \tilde{h}(K_t) \tag{14}$$

$$K_{t+1} = g(K_t, C_t) = {K_t}^{\alpha} + (1-\delta)K_t - C_t \tag{15}$$

という連立方程式を逐次的に解くと、図 3 のようなグラフが描ける。ここで指摘できる重要なポイントは、予め遠い将来に定常状態に至ると仮定しなくとも、$t$ が充分経過した後では、$C_t, K_t$ はそれぞれ定常状態での値 $C^*, K^*$ に一致するという結果である[*10]。しかも、そこに至るまでの経路は、ノート 2 の図 8 に示すのとほぼ同じである。したがって、遠い将来に定常状態に至ると仮定して連立方程式を解くというのは、妥当な方法であったことが（1 つの具体例に過ぎないが）示せた。

---

[*9] ノード数は 20 である（ノードについては後述）。

[*10] その理由の 1 つとして、行動関数 $C_t = \tilde{h}(K_t)$ は点 $(C^*, K^*)$ を通ることが挙げられる。

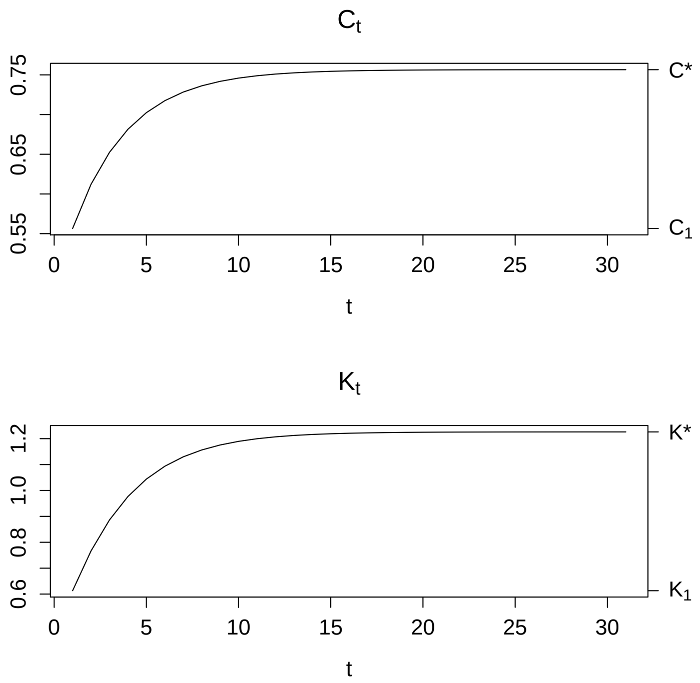


図 3　$C_t, K_t$ の動き $(K_1 = 0.5K^*)$

# 2　数学の準備（５）: 行列 ♠

3 節でモデルの再帰的表現の議論をするには行列を用いる必要があるため、この節で定義と使い方について説明しておく。

## 2.1　行列の定義と演算 ♠

### 行列の定義

自然数 $m, n$ に対し、$mn$ 個の実数 $a_{ij}$ $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ を縦に $m$ 個、横に $n$ 個、長方形状に並べて括弧でくくった表を $(m, n)$ 型の行列という。たとえば、以下の

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \tag{16}$$

は、$(m, n)$ 型の行列である。$A$ の上から $i$ 番目、左から $j$ 番目の要素 $a_{ij}$ を $A$ の $(i, j)$ 成分と呼ぶ。行列の横一列を行といい、行列の縦一列を列という。上から $i$ 番目の行を第 $i$

行といい、左から $j$ 番目の列を第 $j$ 列という。

### 行列の等号

行列 $A, B$ が同じ型の行列であって、対応する成分が全て等しいことを $A, B$ が等しいことと定義し、$A = B$ で表す。

### 行列の和

同じ $(m, n)$ 型の行列 $A, B$ に対し、対応する成分の和を成分とする同じ型の行列を $A, B$ の和と定義し、$A + B$ で表す。すなわち、

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \tag{17}$$

ならば、

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{mm} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{bmatrix} \tag{18}$$

である。

### 行列のスカラー倍

実数 $c \in R$ に対し $(m, n)$ 型の行列 $A$ の各成分を $c$ 倍して得られる同じ型の行列を $cA$ で表す。すなわち、上記の $A$ に対し

$$cA = \begin{bmatrix} c\,a_{11} & c\,a_{12} & \dots & c\,a_{1n} \\ c\,a_{21} & c\,a_{22} & \dots & c\,a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c\,a_{m1} & c\,a_{m2} & \dots & c\,a_{mn} \end{bmatrix} \tag{19}$$

である。

### 行列の積

$A$ が $(l,m)$ 型の行列、$B$ が $(m,n)$ 型の行列の積が、$(l,n)$ 型の行列であってその $(i,k)$ 成分が $\sum_{j=1}^{m} a_{ij}b_{jk}$ で与えられると約束する。すなわち、

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{l1} & a_{l2} & \dots & a_{ln} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \tag{20}$$

に対して、

$$\begin{aligned} AB = C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{l1} & c_{l2} & \dots & c_{ln} \end{bmatrix} \\ c_{ij} = \sum_{j=1}^{m} a_{ij}b_{jk} = a_{i1}b_{1k} + a_{i2}b_{2k} + \cdots + a_{im}b_{mk} \end{aligned} \tag{21}$$

と約束する。一般に $AB$ と $BA$ は等しくならないことに注意しよう（もちろん、等しくなる場合もある)。

### 行列の転置

$(m,n)$ 型の行列 $A$ の縦横を逆にした $(n,m)$ 型の行列を $A$ の転置行列といい、$A^\top$ で表す。

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \tag{22}$$

ならば、

$$A^\top = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \tag{23}$$

である。

### 行列の分割

行列をいくつかの縦線と横線で分割すると、行列は複数の区画に分割される。分けられた各区画は、それ自体が 1 つの行列とみなせる。

例えば、$(3,4)$ 列型の行列 $A$ を

$$A = \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \end{array}\right] \tag{24}$$

と分割する。

$$\begin{aligned} A_{11} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}, \quad A_{12} = \begin{bmatrix} a_{14} \\ a_{24} \end{bmatrix} \\ A_{21} &= \begin{bmatrix} a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad A_{22} = \begin{bmatrix} a_{44} \end{bmatrix} \end{aligned} \tag{25}$$

と定義すると、

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{26}$$

と表せる。

### 正方行列、単位行列

行の数と列の数が等しい行列を正方行列という。特に、$(n,n)$ 型の正方行列であって、その対角成分が全て 1、それ以外が 0 である行列を単位行列といい、$I_n$ で表す。すなわち、

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{27}$$

である。

正方行列 $A$ を $n$ 回掛け合わせたものを $A^n$ と略記する。例えば、$AA$ は $A^2$ と略記する。

### 逆行列

$(n,n)$ 型の正方行列 $A$ に対して

$$XA = AX = I_n \tag{28}$$

となる行列 $X$ が存在するとき、$X$ を $A$ の逆行列という。逆行列の存在する正方行列を正則行列と呼ぶ。正則行列の逆行列 $X$ は唯 1 つ存在し、$A^{-1}$ で表す。$A^{-1}$ を $n$ 回掛け合わせたものを $A^{-n}$ と略記する。

### 列ベクトル、行ベクトル、内積

$(n, 1)$ 型の行列を $n$ 項列ベクトルと呼び、以下では、太字の小文字で表す。例えば、

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}. \tag{29}$$

これを見て分かるように、列ベクトルは縦ベクトルとも呼ばれる。列ベクトルの転置により得られる行列を行ベクトルと呼ぶ。$\boldsymbol{a}^\top$ は行ベクトル

$$\boldsymbol{a}^\top = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} \tag{30}$$

である。行ベクトルは横ベクトルとも呼ばれる。成分が全てゼロの列ベクトルをゼロベクトルと呼ぶ。

2 つの $n$ 項列ベクトル $\boldsymbol{a}, \boldsymbol{b}$ に対して積

$$\boldsymbol{a}^\top \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i \tag{31}$$

を $\boldsymbol{a}$ と $\boldsymbol{b}$ の内積と呼ぶ。実数を成分とする列ベクトル同士の内積は実数である。

### 行列と連立方程式

連立方程式

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned} \tag{32}$$

は、

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{33}$$

すなわち

$$A\boldsymbol{x} = \boldsymbol{b} \tag{34}$$

と行列と列ベクトルにより表現できる。もし、行列 $A$ が正則ならば、(34) 式の両辺に左から $A^{-1}$ を掛けると

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} \tag{35}$$

と $\boldsymbol{x}$ について解ける[*11]。以下の形式で表される $n$ 元連立方程式

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ &\ldots \\ &a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{aligned} \tag{36}$$

も同様に行列と列ベクトルにより表現できる。

## 2.2 行列の対角化 ♠

### 固有値、固有ベクトル

$(n, n)$ 型の正方行列 $A$ に対して、

$$A\boldsymbol{x} = \alpha\boldsymbol{x}, \quad \alpha \in R \tag{37}$$

となる $n$ 項列ベクトル $\boldsymbol{x}$ であって、ゼロベクトルでないものを、$A$ の固有ベクトルといい、$\alpha$ を固有値という[*12]。

固有ベクトルは、スカラー倍（ゼロを除く）しても固有ベクトルなので、この性質を用いて、固有ベクトルはそれ自体との内積が 1 となるように（つまり絶対値が 1 となるように）規格化できる。以下では、固有ベクトルはこのように規格化してあると約束する。

### 行列の対角化

ある $(n, n)$ 型の正方行列 $A$ に対して、$n$ 個の線型独立な固有ベクトルが存在するとする[*13]。$A$ の $n$ 個の固有値 $\alpha_1, \alpha_2, \ldots, \alpha_n$ を対角線上に並べ他の成分をゼロとした $(n, n)$ 型の正方行列を $V$ とする。すなわち、

$$V = \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ 0 & \alpha_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_n \end{bmatrix} \tag{38}$$

[*11] 概念的にそう書けるという意味であって、現実に逆行列を計算する必要は必ずしもない。プログラム言語を用いる場合、多くは (32) 式のような方程式（線型の方程式という）を解くための関数が用意されている。一般に、逆行列の計算には必要な工程が多く、できるべき避けるべきとされる。

[*12] 固有値、固有ベクトルの求め方については、線型代数の教科書を参照していただきたい。（例えば、『線型代数入門』（文献 [10]））。一般的に行列を扱えるプログラム言語には、固有値、固有ベクトルを求める関数が用意されている。

[*13] $(n, n)$ 型の正方行列には、重解も含めて $n$ 個の固有値が存在する。固有値は虚数（実数と 2 乗した値が負となる数との和）となる場合があり、(37) 式の $\alpha$ は虚数でもよい。この重解も含めた $n$ 個の固有値に対して、$n$ 個の固有ベクトルが互いに線型独立になるように上手く選べたとここでは仮定する。

とする[*14]。また、対応する固有ベクトル $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ を横に並べて作った $(n,n)$ 型の正方行列を $S$ とする。すなわち、

$$S = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix}. \tag{39}$$

このとき、

$$\begin{aligned} &\begin{bmatrix} A\boldsymbol{x}_1 & A\boldsymbol{x}_2 & \ldots & A\boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} \alpha_1\boldsymbol{x}_1 & \alpha_2\boldsymbol{x}_2 & \ldots & \alpha_n\boldsymbol{x}_n \end{bmatrix} \\ \Leftrightarrow\ & A\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_n \end{bmatrix} \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ 0 & \alpha_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_n \end{bmatrix} \\ \Leftrightarrow\ & AS = SV \end{aligned} \tag{40}$$

である。(40) 式の両辺に左から $S^{-1}$ を掛けると、

$$S^{-1}AS = V \tag{41}$$

という表現が得られる。これを行列の対角化という。この結果をすぐ次の節で用いる。

# 3　モデルの再帰的表現 ♠

## 3.1　Blanchard and Kahn の合理的期待均衡解の導出方法 ♠

それでは、最適化問題 (1) として定式化されるモデルを近似により再帰的に表現する方法を説明していこう。ただし、ジャンプ変数、状態変数の一方もしくは両方がベクトルである一般の場合を考える。近似によって得られる解を、慣例にならって、合理的期待均衡解と呼ぶことにする[*15]。いくつかの方法で合理的期待均衡解を求めることができるが、そのうち最も基本的な Blanchard and Kahn [1980] の方法を紹介する[*16]。

そのため、最初に（1）最適化の一階条件を求め、モデルを定常状態周りで対数線型近似する。ラグランジュの未定乗数法を用いて最適化の条件を求めても、ベルマン方程式 (3) を最適化の条件を求めても、全く同じ一階の条件式が導出できる[*17]。これは、ジャンプ変数、状態変数の一方もしくは両方がベクトルの場合（例えば RBC モデル）でも同様

*14 このとき、どの順序で固有値を対角線上に並べるかには任意性がある。

*15 これに対して、これまで用いてきた遠い将来に定常状態に至ると仮定して方程式を解く方法は、“完全予見解（perfect foresight solution）” と呼ぶことができる。両者の違いについては、5 節で説明する。

*16 Blanchard, O.J. and C.M. Kahn (1980) “The Solution of Linear Difference Models under Rational Expectations,” Econometrica, pp. 1305-1311.

*17 にもかかわらず、ベルマン方程式を紹介した理由は、ラグランジュの未定乗数法からは行動関数の存在があらわにならないからである。

である。これら一階の条件式、$s_{t+1} = g(s_t, x_t)$ のような遷移式、および生産関数のような同時点間の変数の関係を表す式などからなるモデル方程式を書き出した上で、各内生変数の定常状態を求め、その周りで対数線型近似する。

このように対数線型化されたモデルは、定常均衡値からの乖離として変数変換した $n_x$ 個のジャンプ変数 $\hat{\boldsymbol{x}}_t$ と $n_s$ 個の状態変数 $\hat{\boldsymbol{s}}_t$ によって、

$$B \begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = C \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \tag{42}$$

と表すことができる。もちろん、方程式の数と内生変数（ジャンプ変数 $\hat{\boldsymbol{x}}_t$ と状態変数 $\hat{\boldsymbol{s}}_t$ からなる）の数は等しくなければならない。したがって、$B, C$ は正方行列でなければならない。

この連立方程式を

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = D \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \tag{43}$$

という形式に変形したものが、ここでいうモデルの再帰的表現である。この形に変形するには単純に $B^{-1}$ を左から掛ければよいようにも思われるが（ただし $B$ に逆行列が存在する場合に限る)、たとえそれが可能でもこれだと $\begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix}$ が発散してしまう。つまり最適化問題 (1) を解いたことにはならない。ここで、行動関数が $(n_x, n_s)$ 型の行列 $P$ を用いて

$$\hat{\boldsymbol{x}}_t = P \hat{\boldsymbol{s}}_t \tag{44}$$

と表されると仮定して、モデルが発散しないという条件を用いて (42) 式を解くのが 1.3 節のボックス内でいう（2）にあたる*18。

まず、$A = C^{-1}B$ の $n_x + n_s$ 個の固有値を求める。$n_x + n_s$ 個の固有値の並べ方には任意性があるが、絶対値が小さい順に $\theta_1, \theta_2, \ldots, \theta_{n_x+n_s}$ と並べ、対角行列

$$V = \begin{bmatrix} \theta_1 & 0 & \ldots & 0 \\ 0 & \theta_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \theta_{n_x+n_s} \end{bmatrix} \tag{45}$$

を作る*19。次に、各固有値に対応する固有ベクトルから (39) 式のように正方行列 $S$ を作

*18 $\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{s}}_t$ が定常均衡値からの乖離であることから、定常状態では (44) 式は両辺ともゼロベクトルになることは自明だろう。

*19 $a$ の絶対値とは（$a$ は実数とする)、$a$ が正値または 0 ならばそれ自身の値、$a$ が負値ならば負号を去った値をいう。ゼロの固有値がある場合は、ごく小さい値 $\epsilon$ で置き換えておき、後で $n \to \infty$ の極限をとるとき、$\epsilon \to 0$ と極限をとることにする。

り $Q = S^\top$ と定義すると

$$QAQ^{-1} = V = \begin{bmatrix} \theta_1 & 0 & \dots & 0 \\ 0 & \theta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \theta_{n_x+n_s} \end{bmatrix} \tag{46}$$

と対角化できる。(42) 式は

$$\begin{aligned} & B \begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = C \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & C^{-1} B \begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & Q^{-1} V Q \begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & Q \begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = V^{-1} Q \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \end{aligned} \tag{47}$$

と変形できる。

Blanchard and Kahn [1980] によると、ジャンプ変数に対応する固有値 $\theta_1, \theta_2, \dots, \theta_{n_x}$ の絶対値が全て 1 より小さく、状態変数に対応する固有値 $\theta_{n_x+1}, \theta_{n_x+2}, \dots, \theta_{n_x+n_s}$ の絶対値が全て 1 以上の場合、唯一の解が存在する。ここでは、この条件が満たされていると仮定する。

$$V_x = \begin{bmatrix} \theta_1 & 0 & \dots & 0 \\ 0 & \theta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \theta_{n_x} \end{bmatrix} \tag{48}$$

$$V_s = \begin{bmatrix} \theta_{n_x+1} & 0 & \dots & 0 \\ 0 & \theta_{n_x+2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \theta_{n_x+n_s} \end{bmatrix} \tag{49}$$

と定義し、(47) 式を再帰的に用いると、

$$\begin{aligned} & Q \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = V^{-n} Q \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = Q^{-1} V^{-n} Q \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = Q^{-1} \begin{bmatrix} V_x^{-n} & \mathbf{0} \\ \mathbf{0} & V_s^{-n} \end{bmatrix} \begin{bmatrix} Q_A & Q_B \\ Q_C & Q_D \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \\ \Leftrightarrow\ & \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = Q^{-1} \begin{bmatrix} V_x^{-n}(Q_A \hat{\boldsymbol{x}}_t + Q_B \hat{\boldsymbol{s}}_t) \\ V_s^{-n}(Q_C \hat{\boldsymbol{x}}_t + Q_D \hat{\boldsymbol{s}}_t) \end{bmatrix} \end{aligned} \tag{50}$$

となる。いま、ジャンプ変数に対応する固有値 $\theta_1, \theta_2, \ldots, \theta_{n_x}$ の絶対値が全て 1 より小さいと仮定しているので、$n \to \infty$ のとき $V_x^{-n}$ の対角成分は $\pm\infty$ に発散する。行動関数が (44) 式のように表されるという仮定の下で、$\begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix}$ が発散しないための必要十分条件は、

$$Q_A \hat{\boldsymbol{x}}_t + Q_B \hat{\boldsymbol{s}}_t = \mathbf{0} \tag{51}$$

である[*20]。$Q_A$ は正方行列なので、$Q_B \hat{\boldsymbol{s}}_t$ を移項して左から $Q_A^{-1}$ をかけると

$$\hat{\boldsymbol{x}}_t = -Q_A^{-1} Q_B \hat{\boldsymbol{s}}_t. \tag{52}$$

つまり、$P = -Q_A^{-1} Q_B$ がこの問題の解である。これで行動関数が求まった。

(50) 式は

$$\begin{aligned} & Q \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = \begin{bmatrix} V_x^{-n}(Q_A \hat{\boldsymbol{x}}_t + Q_B \hat{\boldsymbol{s}}_t) \\ V_s^{-n}(Q_C \hat{\boldsymbol{x}}_t + Q_D \hat{\boldsymbol{s}}_t) \end{bmatrix} \\ \Leftrightarrow\ & \begin{bmatrix} Q_A & Q_B \\ Q_C & Q_D \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{t+n} \\ \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ V_s^{-n}(Q_C \hat{\boldsymbol{x}}_t + Q_D \hat{\boldsymbol{s}}_t) \end{bmatrix} \\ \Leftrightarrow\ & \begin{bmatrix} Q_A \hat{\boldsymbol{x}}_{t+n} + Q_B \hat{\boldsymbol{s}}_{t+n} \\ Q_C \hat{\boldsymbol{x}}_{t+n} + Q_D \hat{\boldsymbol{s}}_{t+n} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ V_s^{-n}(Q_C \hat{\boldsymbol{x}}_t + Q_D \hat{\boldsymbol{s}}_t) \end{bmatrix} \end{aligned} \tag{53}$$

と書き直すことができ、$n = 1$ を代入して下側の等号関係を取り出すと、

$$\begin{aligned} & Q_C \hat{\boldsymbol{x}}_{t+1} + Q_D \hat{\boldsymbol{s}}_{t+1} = V_s^{-1} \left[ Q_C \hat{\boldsymbol{x}}_t + Q_D \hat{\boldsymbol{s}}_t \right] \\ \Leftrightarrow\ & \left[ Q_C P + Q_D \right] \hat{\boldsymbol{s}}_{t+1} = V_s^{-1} \left[ Q_C P + Q_D \right] \hat{\boldsymbol{s}}_t \\ \Leftrightarrow\ & \hat{\boldsymbol{s}}_{t+1} = \left[ Q_C P + Q_D \right]^{-1} V_s^{-1} \left[ Q_C P + Q_D \right] \hat{\boldsymbol{s}}_t \\ \Leftrightarrow\ & \hat{\boldsymbol{s}}_{t+1} = A_A \hat{\boldsymbol{s}}_t \end{aligned} \tag{54}$$

と表現できる。まとめると、(42) 式は

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & P A_A \\ \mathbf{0} & A_A \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} = D \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \tag{55}$$

と解けたことになる。これが、モデル方程式の行列による再帰的表現である[*21]。

---

*20 ある種の横断性条件を用いていることになる。その意味の具体例ついては、脚注*24 を参照。

*21 ♠ ここでの議論からわかるように、1 より小さい固有値の数がジャンプ変数の数より多いと（つまり 1 以下の固有値の数が状態変数の数より少ないと）モデルが発散し、1 より小さい固有値の数がジャンプ変数の数より少ないと（つまり 1 以下の固有値の数が状態変数の数より多いと）解が一意に定まらなくなる。

## 3.2　例：RBC モデルの合理的期待均衡解 ♠

それでは、RBC モデルを例に、具体的に合理的期待均衡解を求めてみよう。そのためには、まずモデルを定常状態周りで対数線型近似する必要があるノート 4 の (8)〜(14) 式として並べた RBC モデルの各方程式を対数線型近似すると、以下のようになる。

$$\hat{w}_t - \hat{c}_t = \gamma \hat{l}_t \tag{56}$$

$$\hat{c}_{t+1} - \hat{c}_t = \beta R^* \hat{R}_{t+1} \tag{57}$$

$$\hat{y}_t = \hat{a}_t + \alpha \hat{k}_t + (1-\alpha)\hat{l}_t \tag{58}$$

$$\hat{w}_t = \hat{a}_t + \alpha \hat{k}_t - \alpha \hat{l}_t \tag{59}$$

$$\frac{R^*}{R^*-1}\hat{R}_t = \hat{a}_t + (\alpha-1)\hat{k}_t + (1-\alpha)\hat{l}_t \tag{60}$$

$$K^*\hat{k}_{t+1} = Y^*\hat{y}_t + (1-\delta)K^*\hat{k}_t - C^*\hat{c}_t \tag{61}$$

$$\hat{a}_{t+1} = \rho \hat{a}_t \tag{62}$$

ただし、$R_t = r_t + 1$ と定義し直し、$e_{t+1} = 0$ とおいた[*22]。ジャンプ変数、状態変数はそれぞれ $\hat{\boldsymbol{x}}_t = [\hat{c}_t\ \hat{l}_t\ \hat{y}_t\ \hat{w}_t\ \hat{R}_t]^\top, \hat{\boldsymbol{s}}_t = [\hat{k}_t\ \hat{a}_t]^\top$ と定義できる（あとで固有値を確認する）。

行列 $B, C$ を

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -\beta R^* & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K^* & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{63}$$

$$C = \begin{bmatrix} 1 & \gamma & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\alpha & -1 & 0 & 0 & \alpha & 1 \\ 0 & -\alpha & 0 & -1 & 0 & \alpha & 1 \\ 0 & 1-\alpha & 0 & 0 & -R^*/(R^*-1) & \alpha-1 & 1 \\ -C^* & 0 & Y^* & 0 & 0 & -(\delta-1)K^* & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho \end{bmatrix} \tag{64}$$

と定義すると、(56)〜(62) 式は、

$$B\begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = C\begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \tag{65}$$

---

[*22] ハット（記号 ^ のこと）つきの変数は、全て定常均衡率からの乖離を表す。具体的に書き出しておくと、$\hat{c}_t = \ln(C_t) - \ln(C^*),\ \hat{l}_t = \ln(L_t) - \ln(L^*),\ \hat{y}_t = \ln(Y_t) - \ln(Y^*),\ \hat{w}_t = \ln(w_t) - \ln(w^*),\ \hat{R}_t = \ln(R_t) - \ln(R^*) = \ln(r_t+1) - \ln(r^*+1),\ \hat{k}_t = \ln(K_t) - \ln(K^*),\ \hat{a}_t = \ln(A_t)$ である。

と書ける。パラメータは、ノート4の設定と同一にしておこう。パラメータを代入した後では、

$$B=\begin{bmatrix}0&0&0&0&0&0&0\\1&0&0&0&-1.025&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&14.30&0\\0&0&0&0&0&0&1\end{bmatrix} \tag{66}$$

$$C=\begin{bmatrix}1&1&0&-1&0&0&0\\1&0&0&0&0&0&0\\0&0.7&-1&0&0&0.30&1\\0&-0.3&0&-1&0&0.30&1\\0&0.7&0&0&-29.49&-0.70&1\\-1.316&0&1.673&0&0&13.94&0\\0&0&0&0&0&0&0.9\end{bmatrix} \tag{67}$$

と、$B, C$ は具体的な数値を成分とする行列になる。$A = C^{-1}B$ の固有値は、$\{0,\ 0,\ 0,\ 0,\ 0.939,\ 1.055,\ 1.111\}$ の7つであり、絶対値が1より小さい固有値の数は5個（ゼロは4重解）でジャンプ変数の数に等しく、絶対値が1以上の固有値の数は2個で状態変数の数に等しい[*23]。

この場合、$V_x^{-1}$ は $n \to \infty$ の極限をとらなくとも発散しているが（脚注*19を参照）、

$$Q_A\hat{\boldsymbol{x}}_t + Q_B\hat{\boldsymbol{s}}_t = \mathbf{0} \tag{51'}$$

が、$\begin{bmatrix}\hat{\boldsymbol{x}}_{t+n}\\ \hat{\boldsymbol{s}}_{t+n}\end{bmatrix}$ が発散しないための必要十分条件であることに変わりはない[*24]。前節の議論に従って計算していくと、

$$P=\begin{bmatrix}0.5212&0.3019\\-0.1701&0.5370\\0.1809&1.3759\\0.3510&0.8389\\-0.0278&0.0467\end{bmatrix} \tag{68}$$

$$A_A=\begin{bmatrix}0.948&0.133\\0.000&0.900\end{bmatrix} \tag{69}$$

---

*23 正確には、4つのゼロのうち1つはほぼゼロ（$-2.5\times10^{-17}$）で、厳密にはゼロではない。

*24 その経済学的な説明は以下のとおりである。もし、資本を $+\infty$ に発散させるようなパスが選ばれたとすると、そのときは消費をもっと増やせば効用が改善する。逆に、資本を $-\infty$ に発散させるようなパスが選ばれることはありえない。その際には途中で生産がゼロになり、効用が $-\infty$ に発散する。

と具体的に求まる。これらを用いると、

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{t+1} \\ \hat{\boldsymbol{s}}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & PA_A \\ \mathbf{0} & A_A \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_t \\ \hat{\boldsymbol{s}}_t \end{bmatrix} \tag{70}$$

という表現が得られる。これが RBC モデルの合理的期待均衡解である。

### 3.3 インパルス応答 ♠

いったんモデルが (70) 式のように表現できると、ノート 4 の図 2 に示したようなインパルス応答を簡単に描くことができる。例えば、$t=0$ において定常状態にあり、(62) 式を対数線型化する前の生産性の遷移式

$$\ln(A_{t+1}) = \rho \ln(A_t) + e_{t+1} \tag{71}$$

で、$t = 1$ において $e_1 = 0.05$ というショックが加わったものとしよう。仮定により $A_0 = 1$ であることを用いると、

$$\hat{a}_1 \ = \ln(A_1) = \rho \ln(1) + e_1 = 0.05 \tag{72}$$

である。誤差項 $e_{t+1}$ はもう 1 つの状態変数 $K_t$ の同じ期の値に影響を与えない、つまり $e_1$ は $K_1$ に影響を与えないので、状態変数の初期値 $\hat{\boldsymbol{s}}_1$ は、

$$\hat{\boldsymbol{s}}_1 = \begin{bmatrix} \hat{k}_1 \\ \hat{a}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.05 \end{bmatrix} \tag{73}$$

と表せる。一方で、ジャンプ変数の初期値 $\hat{\boldsymbol{x}}_1$ は、(68) 式の $P$ を用いて、

$$\hat{\boldsymbol{x}}_1 = P\hat{\boldsymbol{s}}_1 \tag{74}$$

と表せる。したがって、これを初期値として (70) 式を再帰的に用いることにより、任意の $t > 1$ について $\hat{\boldsymbol{x}}_t$, $\hat{\boldsymbol{s}}_t$ を求めることができる。この方法により描いたインパルス応答が、図 4 である。

## 4 行動関数の関数近似 ♠♠

この節では、関数近似した行動関数を繰り返し計算によって求める方法を説明する。関数近似の具体的な方法としてチェビシェフ近似を、繰り返し計算の具体的な方法として行動関数反復法（policy iteration method）を用いることにする。

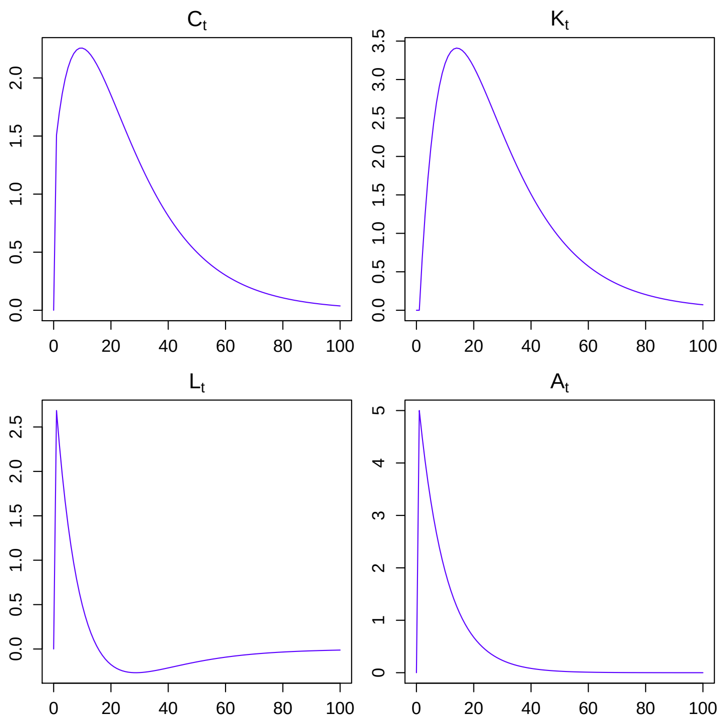


図 4　$e_1 = 0.05$ のショックを与えた場合のインパルス応答、定常均衡値からの乖離率（%）

## 4.1　チェビシェフ近似 ♠♠

関数 $f : R \to R$ の区間 $[a,\ b]$ での $n$ 個の任意の関数 $\hat{\phi}_0, \hat{\phi}_1, \ldots, \hat{\phi}_{n-1}$（基底関数という）を用いた

$$f(x) = \hat{\boldsymbol{\phi}}(x)^\top \boldsymbol{c} = \left[\hat{\phi}_0(x) \quad \hat{\phi}_1(x) \quad \ldots \quad \hat{\phi}_{n-1}(x)\right] \boldsymbol{c} \tag{75}$$

という関数近似を考えよう。ただし、$\boldsymbol{c}$ は $n$ 項列ベクトルで、係数に相当する。一般に、区間 $[a,\ b]$ の全ての点で $f(x) = \hat{\boldsymbol{\phi}}(x)^\top \boldsymbol{c}$ が満たされるようにするためには、ベクトル $\boldsymbol{c}$ の次元が無限でなければならないため、数値計算が不可能である。そこで、基底関数 $\hat{\phi}_0, \hat{\phi}_1, \ldots, \hat{\phi}_{n-1}$ が所与の場合に、区間 $[a,\ b]$ での $n$ 個の点 $x_i,\ i = 1, 2, \ldots, n$ では $f(x_i) = \hat{\phi}(x_i)^\top \boldsymbol{c}$ が成り立つような近似を行う。この $x_i$ をノードといい、一般にノード数と $\boldsymbol{c}$ の次元は一致させる。関数近似をした場合、所定の区間内のノード以外の点では、関数の値を内挿する[*25]。

一般にチェビシェフ近似の場合、ノード数を $n$ とすると

$$z_i = -\cos\left(\frac{i - 0.5}{n}\pi\right), \quad i = 1, 2, \ldots, n \tag{76}$$

*25 内挿は補間ともいう。

をノードとする。$z_i$ は $[-1,\ 1]$ の区間に不均等に並ぶ。基底関数

$$\boldsymbol{\phi}(z) = \begin{bmatrix}\phi_0(z) & \phi_1(z) & \dots & \phi_{n-1}(z)\end{bmatrix}^\top \tag{77}$$

は、

$$\begin{aligned}&\phi_0(z) = 1\\&\phi_1(z) = z\\&\dots\\&\phi_j(z) = 2z\phi_{j-1}(z) - \phi_{j-2}(z)\end{aligned} \tag{78}$$

で与える。チェビシェフ近似のノードと基底関数をグラフに描くと、図 5 のようになる。

いま、$[a,\ b]$ の区間で関数 $V(s), s \in [a,\ b]$ の $V(s) \approx \hat{\phi}(s)^\top \boldsymbol{c}$ というチェビシェフ近似を考える。ノードは

$$g(z) = \frac{(z+1)(b-1)}{2} + a \tag{79}$$

という関数により、上記の $z_i$ を使って $s_i = \gamma(z_i)$ と変換すればよい。

$$z = \gamma^{-1}(s) = \frac{2(s-a)}{b-a} - 1 \tag{80}$$

なので、基底関数は上記の $\phi_j(z)$ を使って、

$$\hat{\phi}_j(s) = \phi_j(\gamma^{-1}(s)), \quad j = 0, 1, 2, \dots, n-1 \tag{81}$$

とおく。つまり、

$$\hat{\boldsymbol{\phi}}(s) = \begin{bmatrix}\phi_0(\gamma^{-1}(s)) & \phi_1(\gamma^{-1}(s)) & \dots & \phi_{n-1}(\gamma^{-1}(s))\end{bmatrix}^\top \tag{82}$$

を基底関数とする。係数 $\boldsymbol{c}$ は、$\boldsymbol{c}$ を未知変数とする $n$ 本の連立方程式

$$v_i = \hat{\boldsymbol{\phi}}(z_i)^\top \boldsymbol{c},\ \ i = 1, 2, \dots, n \tag{83}$$

を解けば求まる。こうして求めた係数 $\boldsymbol{c}$ を用いて

$$\tilde{V}(s) = \hat{\boldsymbol{\phi}}(s)^\top \boldsymbol{c} \tag{84}$$

と定義すれば、関数 $\tilde{V}$ が $V$ のチェビシェフ近似による関数近似となる。

2 変数関数 $f(s_1, s_2)$ をチェビシェフ近似により関数近似する場合、格子状に $n_1 n_2 = n$ 個のノード

$$\boldsymbol{s}_{ij} = [s_i\ s_j]^\top = [\gamma_1(z_i)\ \gamma_2(z_j)]^\top\ \ (i = 1, 2, \dots, n_1,\ j = 1, 2, \dots, n_2) \tag{85}$$

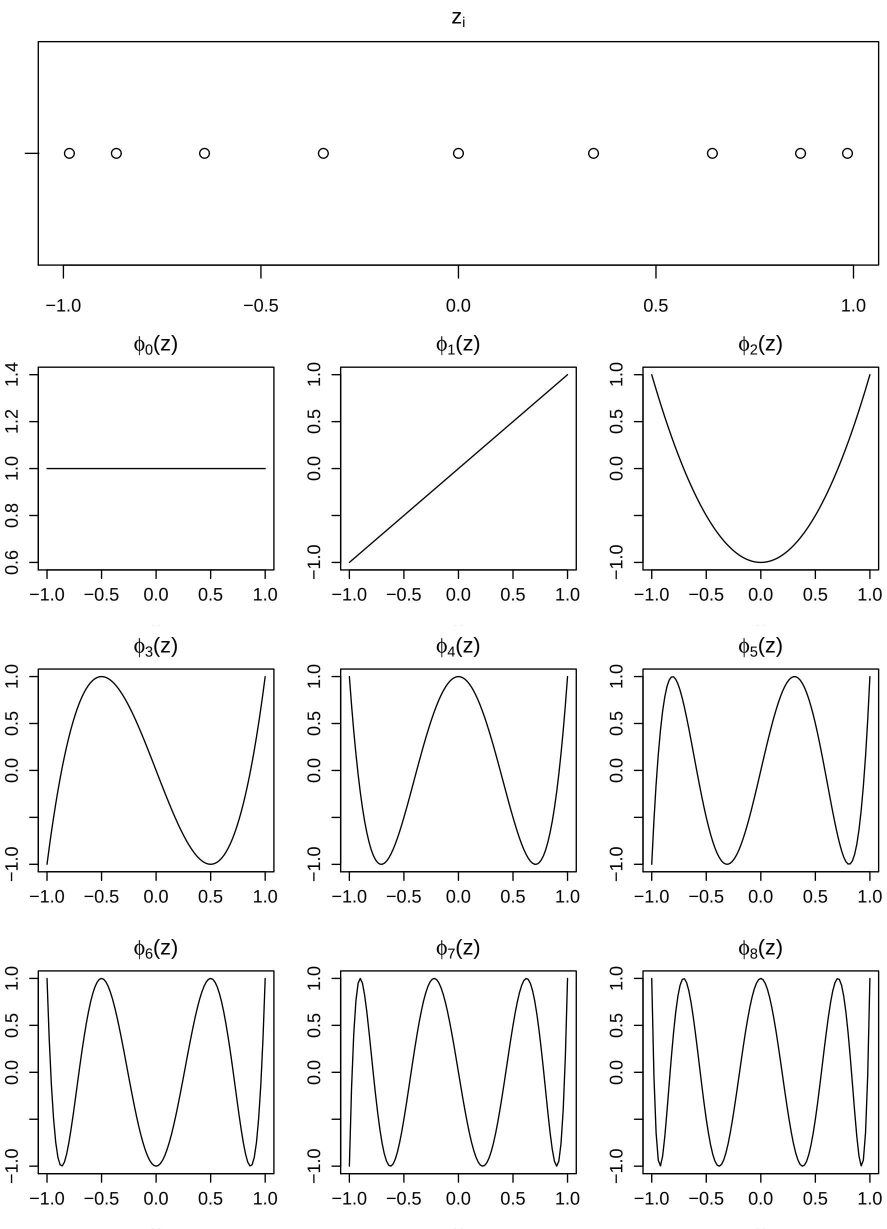


図 5　チェビシェフ近似のノードと基底関数（$n = 9$）

をとり、基底関数は、

$$\hat{\phi}_{ij}(\boldsymbol{s}_{ij}) = \hat{\phi}_j(\gamma_2^{-1}(s_j)) \otimes \hat{\phi}_i(\gamma_1^{-1}(s_i)) \tag{86}$$

とおけばよい。$z_i, z_j$ は (76) 式のように定義し、$\gamma_1, \gamma_2$ は $f$ を関数近似したい領域に応じ

て (79) 式の要領で定義する[*26]。

## 4.2 行動関数反復法（policy iteration method）♠♠

1 節の最適化問題を $n_x$ 個のジャンプ変数と $n_s$ 個の状態変数がある場合に一般化してベルマン方程式を導出しておく。すなわち、

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \sum_{i=0}^{\infty} \beta^i U(\boldsymbol{x}_{t+i}) \\ \text{s.t.} \quad & \boldsymbol{s}_{i+1} = \boldsymbol{g}(\boldsymbol{s}_i, \boldsymbol{x}_i),\ i \geq t \end{aligned} \tag{88}$$

という最適化問題を考える。目的関数 $U$ は $U : R^{n_x} \to R$ であり、$\hat{\boldsymbol{x}}_t$ は $n_x$ 個のジャンプ変数 $x_{1,t}, x_{2,t}, \ldots, x_{n_x,t}$ を縦に並べた列ベクトル、$\hat{\mathbf{s}}_t$ は $n_s$ 個の状態変数 $s_{1,t}, s_{2,t}, \ldots, s_{n_s,t}$ を縦に並べた列ベクトルとする。関数 $\boldsymbol{s}_{t+1} = \boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{x}_t)$ は、それぞれの状態変数に対する遷移式を $g_1, g_2, \ldots, g_{n_s}$ として

$$\boldsymbol{s}_{t+1} = \begin{bmatrix} g_1(\boldsymbol{s}_t, \boldsymbol{x}_t) \\ g_2(\boldsymbol{s}_t, \boldsymbol{x}_t) \\ \vdots \\ g_{n_s}(\boldsymbol{s}_t, \boldsymbol{x}_t) \end{bmatrix} \tag{89}$$

を意味する。

ここで、価値関数 $V(\boldsymbol{s}_t) : \{S_i\}_{i=1}^{n_s} \to R$ を

$$V(\boldsymbol{s}_t) = \max_{\boldsymbol{x}} \sum_{i=0}^{\infty} \beta^i U(\boldsymbol{x}_{t+i}) \tag{90}$$

と定義する。価値関数とは動的計画問題の目的関数のとりうる最大値を状態変数 $\boldsymbol{s}_t$ の関数として表したものである。(90) 式を 1 期先の価値関数 $V(\boldsymbol{s}_{t+1}) = V(\boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{x}_t))$ を用いて、

$$V(\boldsymbol{s}_t) = \max_{\boldsymbol{x}_t} \{U(\boldsymbol{x}_t) + \beta V(\boldsymbol{g}(\boldsymbol{s}_t, \boldsymbol{x}_t))\} \tag{91}$$

と再帰的な形式に書き直したのがベルマン方程式である。既に説明したとおり、ベルマン方程式は数ではなく関数を解とする関数方程式であり、両辺に出てくる関数 $V$ がその解

---

[*26] $\otimes$ はテンソル積（クロネッカー積）で、例えば

$$\begin{bmatrix} 1 \\ b \end{bmatrix} \otimes \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ b \\ a_1 b \\ a_2 b \end{bmatrix} \tag{87}$$

である。

となる関数である。この場合も、ジャンプ変数 $\boldsymbol{x}_t$ のとる値を状態変数 $\boldsymbol{s}_t$ の関数として表した行動関数が定義でき、行動関数は

$$\boldsymbol{x}_t = \boldsymbol{h}(\boldsymbol{s}_t) = \begin{bmatrix} h_1(\boldsymbol{s}_t) \\ h_2(\boldsymbol{s}_t) \\ \vdots \\ h_{n_x}(\boldsymbol{s}_t) \end{bmatrix} \tag{92}$$

と複数の関数を集めたものになる。(4) 式と同様に、行動関数は

$$\boldsymbol{h}(\boldsymbol{s}_t) = \underset{\boldsymbol{x}_t}{\operatorname{argmax}} \{U(\boldsymbol{x}_t) + \beta V\left(g(\boldsymbol{s}_t, \boldsymbol{x}_t)\right)\} \tag{93}$$

と表せる。多くの場合、行動関数 $\boldsymbol{h}$ を解析的に求めることは不可能なので、それを具体的に求めたい場合には、他の関数で近似する必要がある。

関数近似した行動関数は、(93) 式の関係を利用して以下の行動関数反復法というアルゴリズムにより求められる[*27]。

行動関数反復法（Policy Iteration Method）

ノードをセット（ノード数を $n$ とする）
$V$ の関数近似の初期値 $\tilde{V}^{(0)}$ をセット
repeat
　各ノード $\boldsymbol{s}_k$ について $\boldsymbol{x}_t^{(j)} \leftarrow \operatorname{argmax}_{\boldsymbol{x}_t} \left[U(\boldsymbol{x}_t) + \beta\tilde{V}^{(j-1)}(\boldsymbol{g}(\boldsymbol{s}_{kl}, \boldsymbol{x}_t))\right]$ を計算
　$\{\boldsymbol{x}_t^{(j)}\}_{k=1}^{n}$ により関数近似 $\tilde{\boldsymbol{h}}^{(j)}$ を定義
　各ノード $\boldsymbol{s}_k$ について $\boldsymbol{v}^{(j)} \leftarrow \sum_{i=0} \beta^i U(\tilde{\boldsymbol{h}}^{(j)}(\boldsymbol{x}_{t+i}))$ を計算
　$\boldsymbol{v}^{(j)}$ により $\tilde{V}^{(j)}$ を定義
until $|\boldsymbol{v}^{(j)} - \boldsymbol{v}^{(j-1)}| < \epsilon$

チェビシェフ近似により関数近似を行うとすると、まず 4.1 節の要領でノードを決める。$V$ の関数近似の候補から各ノードにおける値を求め、係数 $\boldsymbol{c}^{V(0)}$ を求める。ここまでが、「$V$ の関数近似の初期値 $\tilde{V}^{(0)}$ をセット」にあたる。次に、各ノードについてベルマン方程式を満たす $\boldsymbol{x}_t$ の組み合わせを数値計算により求め、それを元に行動関数 $\boldsymbol{h}$ の関数近似 $\tilde{\boldsymbol{h}}$ を求める。さらに、$\tilde{\boldsymbol{h}}$ を使って各ノードにおける価値関数の値を求め、係数 $\boldsymbol{c}^{V(j)}$ を更新する。ここまでが、繰り返し計算の中身である。

繰り返し計算の中で、全てのノードにおける価値関数 $V$ の値 $\boldsymbol{v}$ が変化しなくなった場合に、アルゴリズムが収束したとみなして、終了する。$\epsilon$ は収束判断の基準で、小さい値

[*27] ここではごく簡単な説明に留める。詳細や他の具体例については、“Applied Computational Economics and Finance”（文献 [5]）を参照していただきたい。

に設定する[*28]。このような収束計算の結果として求めた行動関数の関数近似 $\tilde{\boldsymbol{h}}$ をインパルス応答の計算などに利用する。

## 4.3　例：RBC モデルの行動関数の関数近似 ♠♠

RBC モデルのベルマン方程式は、

$$V(K_t, A_t) = \max_{C_t, L_t} \left\{ \ln(C_t) + \mu L^{\gamma+1} + \beta V\left(K_{t+1}, A_{t+1}\right)\right\} \tag{94}$$

$$K_{t+1} = A_t K_t^{\alpha} L_t^{1-\alpha} + (1-\delta)K_t - C_t \tag{95}$$

$$A_{t+1} = A_t^{\rho} \tag{96}$$

と書ける。本来、$\ln(A_{t+1}) = \rho \ln(A_t) + e_{t+1}$ だが、$e_{t+1}$ は $t$ 時点には知りえないものとして、その平均値である 0 で置き換えた上で、両辺の指数関数をとった[*29]。

具体的に、行動関数反復法により行動関数の関数近似を求めよう。パラメータはノート 4 の設定と同じにし（したがって 3.2 節と同じ）、定常状態では $K^* = 14,3,\ A^* = 1$ であることを考慮して $K$ については区間 [1, 30] でノードをとり、$A$ をついては区間 [0.9, 1.1] でノードをとる。ノード数は、両方とも 15 とした。したがって、$15^2 = 225$ 個の基底関数と係数により価値関数と $C_t$ についての行動関数を

$$V(K_i, A_j) \approx \tilde{V}(K_i, A_j) = \hat{\boldsymbol{\phi}}^V(K_i, A_j)^\top \boldsymbol{c}^V \tag{97}$$

$$h^C(K_i, A_j) \approx \tilde{h}^C(K_i, A_j) = \hat{\boldsymbol{\phi}}^C(K_i, A_j)^\top \boldsymbol{c}^C \tag{98}$$

と関数近似する。$L$ についての行動関数は、一階の条件により

$$L_t = \left\{ \frac{(1-\alpha)A_t K_t^{\alpha}}{(\gamma+1)\mu C_t} \right\}^{\frac{1}{\gamma+\alpha}} \tag{99}$$

で、$t$ 期の状態変数と $C_t$ の関数なので関数近似の必要はない。

行動関数反復法により求めた行動関数の関数近似 $\tilde{h}^C(K_t, A_t)$ をグラフに描くと、図 6 のようになる。$h^C$ が 2 変数関数なので、$\tilde{h}^C$ も空間における曲面として描かれる[*30]。$t = 1$ において $K_1$ は定常均衡値 $K^*$ に一致し、$A_1$ は定常均衡値 1 より 5 ％大きい $A_1 = 1.05$ にあったとしよう。図 4 を描いたときと同様、$t = 0$ において $t = 1$ において $e_1 = 0.05$ というショックが加わった場合とも解釈できる。この方法で求めた行動関数

[*28] $|\boldsymbol{a} - \boldsymbol{b}| = \sqrt{(\boldsymbol{a} - \boldsymbol{b})^\top (\boldsymbol{a} - \boldsymbol{b})}$ という意味で、$|\boldsymbol{v}^{(j)} - \boldsymbol{v}^{(j-1)}| < \epsilon$ とは、大雑把に “$\boldsymbol{v}^{(j)}$ と $\boldsymbol{v}^{(j-1)}$ がほぼ同じこと” を意味する。これが、繰り返し計算の終了の条件である。

[*29] $e_i,\ i \geq t+1$ の分布のみ知っている（例えば独立に平均ゼロ、分散一定の正規分布従う）として計算することもできる。この場合計算量は増えるが、$e_i,\ i \geq t+1$ をゼロで置き換えて計算した場合と一般にはそれほど大きな差は出ない。

[*30] 行動関数を行列 $P$ を用いて $\hat{\boldsymbol{x}}_t = P\hat{\boldsymbol{s}}_t$ と近似する 3 節の方法では、近似した行動関数をグラフに描くと、定常状態を表す点を通る平面になる。行動関数 $C_t = h^C(K_t, A_t)$ 自体も、定常状態を表す点を通る。

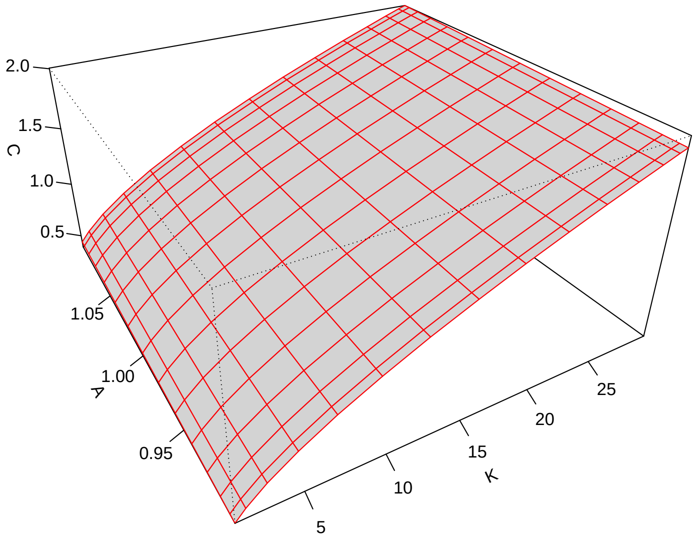


図 6　行動関数 $C_t = h^C(K_t, A_t)$ のチェビシェフ近似（ノード数は $15^2$）

$h^C(K_t, A_t)$ の関数近似 $\tilde{h}^C(K_t, A_t)$ により描いたインパルス応答が、図 7 である。図 4 とほぼ同じであることは明白であろう。

この場合でも、この場合も $t$ を充分大きくとると、$C_t, K_t$ は定常均衡値に近づく。したがって、遠い将来に定常状態に至ると仮定して連立方程式を解くというのは、RBC モデルでも妥当な方法であったことが示せた[*31]。

# 5　最適化手法のまとめ

ここまで、モデルの解、すなわちインパルス応答やシミュレーション結果を求める方法を複数紹介してきたが、それらの特徴を箇条書き形式でまとめておく（図 8 も参照）。

- 最適化の一階条件（a）は、ラグランジアンを用いてもベルマン方程式を用いても全く同じ方程式が導出される。
- 行動関数の関数近似（b）は、繰り返し計算が必要なため計算コストが大きいが、解の範囲に制約がある場合など、ベルマン方程式に微分不可能な点が含まれる場合

*31　ノート 4 でのように遠い将来に定常状態に至ると仮定して連立方程式を解くことによって描いたインパルス応答と図 4、図 7 とがどれだけ近いかは、読者自身でぜひ確かめていただたい。

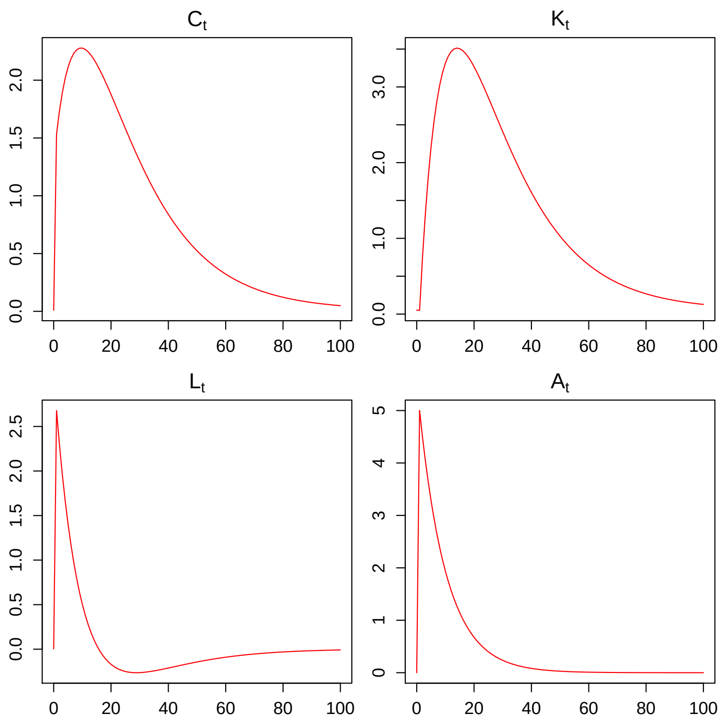


図 7　$e_1 = 0.05$ のショックを与えた場合のインパルス応答、定常均衡値からの乖離率（%）

に、特に有効である。

- 連立方程式として解く場合（c）は、それぞれの主体が将来の外生変数の値を全て知っているという前提で解くことになる（完全予見の仮定）。価値関数・行動関数が毎期変化する場合にも対応できるので、モデルを行列により再帰的表現した場合よりも計算可能なシミュレーションのパターンが多い。
- 線型近似した行動関数によるモデルの再帰的表現（d）は、パラメータ推定に有用である（ノート 7 で説明する）。方法のバリエーションは複数存在する*32。二次近似する方法も知られている。
- VAR（Vector Auto-Regression）とは、来期のモデル変数の値を当期以前のモデル変数の値の関数として再帰的に表現したモデルをいう。線型近似した行動関数によるモデルの再帰的表現（d）による VAR 表現は行列で表現できる線型 VAR であり、反復計算を用いた行動関数の関数近似（b）による VAR 表現は行列で表現できない非線型 VAR になる。

*32 最も基本的なのが Blanchard and Kahn の方法だが、他にも Sims の方法など。
Sims, C. A. (2002) “Solving Linear Rational Expectations Models,” Computational Economics, Vol. 20, No. 1-2, pp. 1-20.

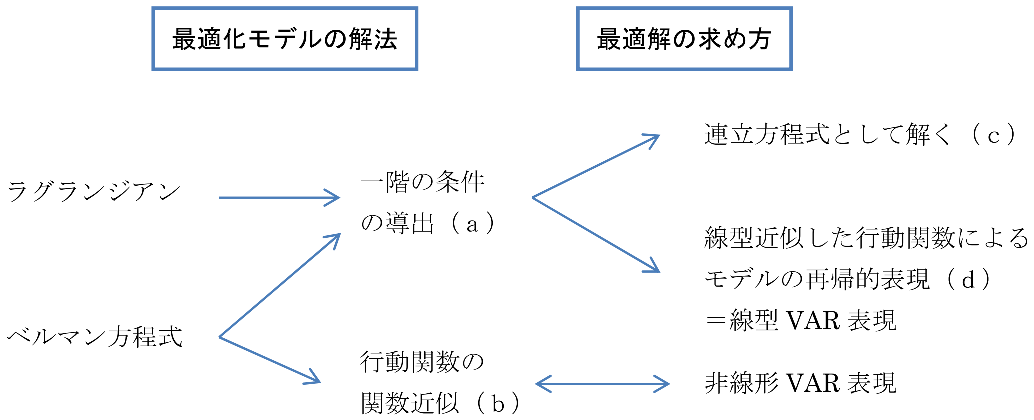


図 8　解法の比較

ラグランジュの未定乗数法は単に極値を求める方法であって、厳密には、本当に最適化が実現されているかどうか他の方法によって確かめる必要がある。最適成長モデル、RBC モデルでは、ベルマン方程式から解析的に最適解のパスの存在可能性を議論することは可能であるが、ここではあえてそのような数学的な議論は避けて、近似計算によって最適解の性質、特に、どのような初期値が与えられても（途中に何もショックがなければ）遠い将来に定常状態に到達することを数値計算により示した*33。ただし、実用上は、モデルの解が一意に存在するかの確認には、モデルの再帰的表現（d）の途中で得られる行列の固有値の絶対値を調べるのが簡単でよい（脚注*21 も参照）。

ニューケインジアン・モデルでは、異時点間の最適化問題が家計の効用最大化問題と企業の利潤の割引現在価値最大化問題の 2 つ存在する。この場合、複数の価値関数が存在するため、前節までのような 1 つのベルマン方程式を用いた議論はそのまま適用できない。ただし、ニューケインジアン・モデルでは、一階の最適化条件と状態変数の遷移式を合わせるとちょうど未知変数の数と一致するため、いわば競争均衡解と呼ぶべき解が求まる。

---

*33 動的計画法のより数学的に厳密な議論に関心のある読者は、“Introduction to Modern Economic Growth”（文献 [1]）を参照していただきたい。
（章末脚注）本章の記述の多くは、『現代マクロ経済学講義』（文献 [8]）と “Applied Computational Economics and Finance”（文献 [5]）を参考にした。DSGE モデルについてより深く学びたい読者は前者に、動的計画法に興味を持った読者は後者に進んでいただきたい。チェビシェフ近似については結果を紹介するに留めたが、フーリエ級数展開との関係など数学的な意味については『数値解析』（森正武著）が詳しい。